Technical Updates: Solitaire: Man Versus Machine

Xiang Yan

Persi Diaconis

Paat Rusmevichientong

Benjamin Van Roy

Stanford University

Xyan,persi.diaconis,bvr@stanford.edu

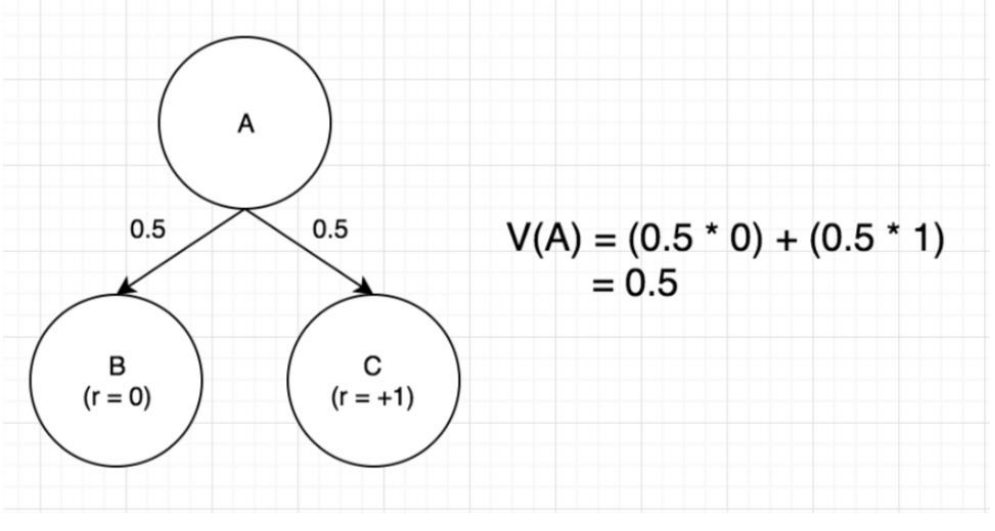
Cornell University
paatrus@orie.cornell.edu

1 Technical Updates

We have used a rollout method to compute the chances of winning and implementing the techniques. Rollout method uses the current strategy to update the same. So, one possible direction is to compute a value function.

Value Function:

The value function calculates the prospective future benefits of existing in this state.



We attempted to find the value of state A in the diagram. There's a 50-50 chance you'll end up in one of the next two states, B or C. The value of state A is equal to the total of the probabilities of all subsequent states multiplied by the incentive for reaching that state. As a result, state A has a value of 0.5.

So, at any progression step other than the terminal stage, the agent does an action that leads to the next state, which may or may not result in a reward, but will bring the agent closer to receiving a reward.

The worth of the algorithm for determining the value of existing in a state, or the likelihood of obtaining a future reward, is known as the function. The value of each state is updated in reverse chronological order throughout a game's state history; given enough training, the agent will be able to determine the true value of each state in the game using both explore and exploit strategies.

Rollout Method:

As discussed in the paper rollout method is based on the following procedure:

1. For each legal move a, simulate the remainder of the game, taking move a and then employing strategy h thereafter

2. If any of these simulations leads to victory, chose one of them randomly and let h'(x) be the corresponding move a.

3. If none of the simulations lead to victory, let h'(x)=h(x).

This procedure can further generate a further improved strategy h'' that is a rollout strategy relative to h'. So, after a finite number of such iteration, we can arrive at an optimal strategy.

We can see that the value function is a further extension of the rollout method. But value function doesn't disregard the outcome which does not take it to the desired result instead it keeps it as a memory and use it in further stages.

Certainly, this function could not represent exactly, but we could try approximating it in terms of a linear combination of features of the game state.

Moreover, one thing that can be further improved is the speed and time complexity of the algorithm Currently, the tightest upper bound we can rigorously prove is 98.81%. As discussed in the paper we cannot use this algorithm for the game in which there is a time limit. So, modifying this algorithm in such a manner that it can be optimized in every mode of the game would be a great update.

If the success rate bound is improved and we are able to run additional rollout iteration, we may produce a verifiable near-optimal strategy for thoughtful solitaire.

References:

Value Function - https://towardsdatascience.com/reinforcement-learning-value-function-57b04e911152

Rollout - https://papers.nips.cc/paper/2004/file/48c3ec5c3a93a9e294a8a6392ccedeb4-Paper.pdf